UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,033	09/18/2003	Cary L. Bates	ROC920030169US1	7930

46797 7590 02/07/2007
IBM CORPORATION, INTELLECTUAL PROPERTY LAW
DEPT 917, BLDG. 006-1
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
MITCHELL, JASON D	
ART UNIT	PAPER NUMBER
2193	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.**	**Applicant(s)**	
	10/666,033	BATES ET AL.	
	Examiner	**Art Unit**	
	Jason Mitchell	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 9/18/03.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-41 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-41 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 18 September 2003 is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 9/18/03.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Claims 1-41 are pending in this application.

Claim Objections

2. **Applicant is advised that should claims 14-18 be found allowable, claims 19 and 23-26 will be objected to under 37 CFR 1.75 as being a substantial duplicate thereof.** When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See MPEP § 706.03(k).

Claims 14 and 19 respectively recite the selected target call site comprising "a routine" and "a method". Applicant has failed to make any significant distinction between the two terms and in fact has indicated that they are synonymous (see par. [0009] 'sets of instructions ... referred to as "routines" or "methods"').

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first and second paragraphs of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. **Claims 9, 14, 19, 28 and 33 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement.** The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 14 is exemplary and recites, “call context information uniquely identifying the selected target call site”.

In pars. [0055]-[0056] of the specification, Applicant states, in part:

> Unambiguous identification of the call site may be necessary, for example, where two or more objects are objects of the same class. Consider the following code: foo(obj1.bar(), obj2.bar()), where obj1 and obj2 are of the same class. Now assume the user desires to run into obj2.bar(). Without unique identification of the call site, execution may halt at obj1.bar().
>
> The particular technique for unambiguously identifying the call site may vary depending on the particular implementation. In one embodiment, the line number of the call site is known and is mapped to information (i.e., the "calling context information") that will appear on the call stack 120, e.g., a statement number or address.

While Applicant discloses a need to ‘unambiguously’ identify a call site, the only enabled identification method (i.e. recording the ‘line number’ of the call site) does not address this need. Specifically, in the given example code both call sites (i.e. ‘obj1.bar()’ and ‘obj2.bar()’) are on the same line and consequently, a line number does not between the two calls.

5. **Claims 14, 19, and 33 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and**

distinctly claim the subject matter which applicant regards as the invention.

The claims variously recite a “target call site comprising a routine having a plurality of entry points” and a “target call site comprising a method”. The specification indicates that a target call site represents “the location from which the function of interest is called” (par. [0031]). Thus, the call site must refer to a routine or method of interest and cannot comprise that routine or method of interest.

For the purposes of examination the claims will be treated as reading “the target call site comprising a call to” the appropriate routine or method.

6. **Claims 11, 16, 24, 30 and 35 recite the limitation "the stored call context information". There is insufficient antecedent basis for this limitation in the claim.** Specifically, the parent claims have ‘determined’ call context information but there is no explicit recitation of storing it.

7. **Claim 15 recites the limitation "the at least one associated breakpoint", in line 5. There is insufficient antecedent basis for this limitation in the claim.** This would appear to refer to the ‘plurality of associated breakpoints’, however it is unclear if applicant intends to refer to all of the plurality or merely a subset.

8. **Claim 19 and 33 recite the limitation "the routine". There is insufficient antecedent basis for this limitation in the claim.** The claims are directed to call sites comprising 'methods' instead of 'routines'.

Claim Rejections - 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. **Claims 27 and 33 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.**

11. The claims are not limited to tangible embodiments. In view of Applicant's disclosure in par. [0032] the medium is not limited to tangible embodiments, instead being defined as including both tangible embodiments (e.g., "storage media") and intangible embodiments (e.g., "communications" media). As such, the claim is not limited to statutory subject matter and is therefore non-statutory.

Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

13. **Claims 1-2 and 38-39 are rejected under 35 U.S.C. 102(b) as being anticipated by US 6,077,312 to Bates et al. (Bates).**

14. **Regarding Claim 1:** Bates discloses a computer-implemented method for debugging code, comprising:

receiving a selection of a target call site, the target call site comprising a call to a routine (col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made"; col. 7, lines 20-23 "a user configures the ... context sensitive breakpoint"); and

setting at least one run into breakpoint configured to halt subsequent execution (col. 5, lines 61-65 "context sensitive breakpoints that conditionally halt execution") only when an execution path arrives at the routine from the target call site (col. 6, lines 36-39 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made").

15. **Regarding Claim 2:** The rejection of claim 1 is incorporated; further Bates discloses:

the selection is received while execution is halted at a first point in the code (Fig. 5, steps 80 and 82);

the target call site is located at a second point in code (Fig. 5, steps 96 and 106); and

the first point is a first statement in the code (Fig. 13, step 226) and the second point is a second statement in the code (Fig. 5, steps 96 and 106).

16. **Regarding Claim 38:** Bates discloses a computer, comprising:

a memory (col. 4, line 15 "memory");

code under debug resident in the memory (col. 5, line 11 "executable program 42"), the code comprising as least one target call site comprising a call to a routine selected by a user (col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made"; col. 7, lines 20-23 "a user configures the ... context sensitive breakpoint");

a breakpoint data structure resident in the memory and configured for storing at least context information indicating a location of the call within the code (col. 12, Table I, "Exact Match ... [routineName,statemetnID]"; Fig. 10, Breakpoint Table 62); and

a debugger program resident in the memory and configured to interrupt execution of the code under debug in response to encountering a breakpoint set on the routine if the routine is called from the target call site as determined with reference to the context information (col. 6, lines 24-29 "upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria").

17. **Regarding Claim 39:** The rejection of claim 38 is incorporated; further Bates discloses a caller data structure resident in the memory and configured for storing at least callers of routines in the code as encountered during an execution

path (col. 6, lines 24-29 "calling stack"); and wherein the debugger program is configured to determine whether the routine is called from the target call site by comparing the context information to a caller stored in the caller data structure (col. 6, lines 24-29 "determine whether the calling stack matches the predetermined criteria").

Claim Rejections - 35 USC § 103

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

19. **Claims 3 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,077,312 to Bates et al. (Bates).**

20. **Regarding Claims 3 and 40:** The rejections of claims 1, 19 and 38 are incorporated, respectively; further it would at least have been obvious to one of ordinary skill in the art to apply the Bates invention to object-oriented code wherein the target call site is a method of the object-oriented code because "the invention addresses these and other problems" (col. 2, lines 40-45) which are "especially pronounced in object-oriented programming" (col. 1, line 66-col. 2, line 3).

21. **Claims 4-18, 21-37 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,077,312 to Bates et al. (Bates) in view of US 2003/01067046 to Arnold et al. (Arnold).**

22. **Regarding Claim 4:** The rejection of claim 1 is incorporated; further Bates does not discloses setting the run into breakpoint on an instruction calling the routine.

23. Arnold teaches setting the at least one run into breakpoint comprises setting the run into breakpoint on an instruction calling the routine (pg. 7, claim 11 "setting a breakpoint on a method call to an implementation of the method").

24. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates and Arnold in order to provide an improved method for debugging object oriented programs (Bates col. 1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming"; Arnold par. [0027] "facilitate the debugging of object-oriented computer programs").

25. **Regarding Claim 5:** The rejection of claim 1 is incorporated; further Bates does not disclose setting the run into breakpoint at each entry point to the routine.

26. Arnold teaches setting the at least one run into breakpoint comprises setting the run into breakpoint at each entry point to the routine (par. [0027] "halt execution ... in response to reaching any of a plurality of implementation of a method"; also see claim 10 on pg. 7 "setting a breakpoint on a first statement in an implementation of the method").

27. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates and Arnold in order to provide an improved method for debugging object oriented programs (Bates col. 1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming"; Arnold par. [0027] "facilitate the debugging of object-oriented computer programs").

28. **Regarding Claim 6:** The rejection of claim 5 is incorporated; further Bates discloses:

encountering the at least one run into breakpoint (col. 15, lines 16-19 "breakpoint handler module 196 is invoked to handle the breakpoint");

determining whether the routine is entered from the selected target call site (col. 15, lines 21-23 "starts execution of the ... test breakpoint routine"); and if so, halting execution of the code (col. 15, lines 23-28 "should it be determined that the breakpoint is triggered ... halts execution").

29. **Regarding Claims 7 and 27:** Bates discloses a computer-implemented method for debugging code, comprising:

receiving a selection of a target call site in the code, the target call site comprising a call to a routine (col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made"; col. 7, lines 20-23 "a user configures the ... context sensitive breakpoint");

determining call context information identifying a location of the selected target call site in the code (col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made");

upon encountering the breakpoint during execution of the code, determining whether the routine is called from the selected target call site based on the call context information (col. 6, lines 24-29 "upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria"); and

if so, halting execution of the code (col. 15, lines 23-28 "should it be determined that the breakpoint is triggered ... halts execution").

30. Bates does not disclose establishing a breakpoint at an entry point to the selected target call site.

31. Arnold teaches establishing a breakpoint at an entry point to the selected target call site (par. [0027] "halt execution ... in response to reaching any of a plurality of implementation of a method"; also see claim 10 on pg. 7 "setting a breakpoint on a first statement in an implementation of the method");

32. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates and Arnold in order to provide an improved method for debugging object oriented programs (Bates col. 1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming"; Arnold par. [0027] "facilitate the debugging of object-oriented computer programs")

33. **Regarding Claim 8:** The rejection of claim 7 is incorporated; further Bates discloses the target call site is located at a first point in the code (Fig. 5, steps 96 and 106) and the selection of the target call site is received while execution is halted at a second point in the code (Fig. 5, steps 80 and 82).

34. **Regarding Claims 9 and 28:** The rejections of claims 7 and 27 are incorporated, respectively; further Bates discloses the call context information unambiguously identifies the code (col. 12, Table I, "Exact Match ... [routineName,statemetnID]").

35. **Regarding Claims 10 and 29:** The rejections of claims 7 and 27 are incorporated, respectively; further Bates implies the code is object-oriented and the target call site is a method of the object-oriented code (col. 1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming").

36. **Regarding Claims 11 and 30:** The rejections of claims 7 and 27 are incorporated, respectively; further Bates discloses determining whether the routine is called from the selected target call site based on the stored call context information comprises comparing the stored call context information to selected content of a call stack (col. 6, lines 24-29 "a test ... to determine whether the calling stack matches the predetermined criteria").

37. **Regarding Claims 12 and 31:** The rejections of claims 11 and 30 are incorporated, respectively; further Bates discloses the routine is determined to have been called from the selected target call site if the stored call context information matches the selected content of the call stack (col. 6, lines 24-29 "determine whether the calling stack matches the predetermined criteria").

38. **Regarding Claims 13 and 32:** The rejections of claims 11 and 30 are incorporated, respectively; further Bates discloses the selected content of the call stack is a call to the routine and wherein the routine is determined to have been called from the selected target call site if the stored call context information matches the selected content of the call stack (col. 6, lines 24-29 "determine

whether the calling stack matches the predetermined criteria"; col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made").

39. **Regarding Claims 14, 19 and 33:** Bates discloses a computer-implemented method for debugging code, comprising:

(a) receiving a selection of a target call site in the code, the target call site comprising a routine having a plurality of entry points (col. 6, lines 36-41 "the predetermined criteria also compares the specific instructions in such routines from which routine calls are made"; col. 7, lines 20-23 "a user configures the ... context sensitive breakpoint");

(c) determining call context information uniquely identifying the selected target call site (col. 12, Table I, "Exact Match ... [routineName,statemetnID]"); and

(d) for each of the breakpoints encountered during execution of the code:

determining whether the routine is called from the selected target call site based on the call context information (col. 6, lines 24-29 "upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria"); and

if so, halting execution of the code (col. 15, lines 23-28 "should it be determined that the breakpoint is triggered ... halts execution").

40. Bates does not disclose (b) establishing a breakpoint at each of the plurality of entry points;

41. Arnold teaches (b) establishing a breakpoint at each of the plurality of entry points (par. [0027] "halt execution ... in response to reaching any of a plurality of implementation of a method"; also see claim 10 on pg. 7 "setting a breakpoint on a first statement in an implementation of the method").

42. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates and Arnold in order to provide an improved method for debugging object oriented programs (Bates col. 1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming"; Arnold par. [0027] "facilitate the debugging of object-oriented computer programs").

43. **Regarding Claims 16, 24 and 35:** The rejections of claims 14, 19 and 33 are incorporated, respectively; further Bates discloses determining whether the routine is called from the selected target call site based on the stored call context information comprises comparing the stored call context information to selected content of a call stack (col. 6, lines 24-29 "upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria").

44. **Regarding Claims 17, 25 and 36:** The rejections of claims 16, 24 and 35 are incorporated, respectively; further Bates discloses the breakpoint is determined to be called from the selected target call site if the stored call context information matches the selected content of the call stack (col. 6, lines 24-29 “upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria”).

45. **Regarding Claims 18, 26 and 37:** The rejections of claims 16, 24 and 35 are incorporated, respectively; further Bates discloses the selected content of the call stack is a call to the routine and wherein the routine is determined to be called from the selected target call site if the stored call context information matches the selected content of the call stack (col. 6, lines 24-29 “upon reaching the breakpoint during execution ... determine whether the calling stack matches the predetermined criteria”; col. 6, lines 36-41 “the predetermined criteria also compares the specific instructions in such routines from which routine calls are made”).

46. **Regarding Claim 20:** The rejection of claim 19 is incorporated; further Arnold teaches prior to identifying, determining that the method involves an object (see Fig. 5, step 116 handles objects (i.e. classes) differently that interfaces (see step 126) and thus must have made the determination prior to identification (step 122)).

47. **Regarding Claim 21:** The rejection of claim 19 is incorporated; further Arnold teaches identifying comprises traversing a class hierarchy (par. [0059] "obtain a ... class hierarchy"; [0060] "Next, a loop is initiated in block 116 to process each class identified").

48. **Regarding Claim 22:** The rejection of claim 19 is incorporated; further Arnold teaches identifying comprises traversing a class hierarchy and locating each matching member method according to the selected target call site (par. [0061] "For each such method ... determine whether the method is identified in the inheritance breakpoint table").

49. **Regarding Claim 41:** The rejection of claim 38 is incorporated; further Bates does not disclose the routine of the target call site is a method and is defined for at least two objects.

50. Arnold teaches the routine of the target call site is a method and is defined for at least two objects; and wherein the debugger program is configured to place breakpoints at each entry point to the method (par. [0027] "halt execution ... in response to reaching any of a plurality of implementation of a method").

51. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates and Arnold in order to provide an improved method for debugging object oriented programs (Bates col.

1, line 66-col. 2, line 3 "This problem is especially pronounced in object-oriented programming"; Arnold par. [0027] "facilitate the debugging of object-oriented computer programs").

52. **Claims 15, 23 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,077,312 to Bates et al. (Bates) in view of US 2003/01067046 to Arnold et al. (Arnold) further in view of US 5,050,168 to Paterson (Paterson).**

53. **Regarding Claims 15, 23 and 34:** The rejections of claims 14, 19 and 33 are incorporated, respectively; further Bates discloses:

repeating each of the steps (a)-(d) for a plurality of target call sites each having a routine with a plurality of entry points, so that each target call site has a plurality of associated breakpoints set at each of the associated plurality of entry points (Fig. 5, note the path from step 110 to step 84); and

automatically removing only the at least one associated breakpoint associated with a particular selected target call site upon determining that the routine is called from the particular selected target call site based on the call context information.

54. Paterson teaches that 'temporary' breakpoint were known in the art at the time of the invention, and further teaches upon encountering such a breakpoint

automatically removing it (col. 7, lines 22-27 “this breakpoint will be removed from the target program if it is later encountered”)

55. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Bates-Arnold and Paterson because the Bates states that “Additional events may also be supported, e.g., removing existing breakpoints” (col. 9, lines 55-58).

Conclusion

56. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

57. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

58. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

59. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jason Mitchell
2/2/07

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100